P&S Modern SSDs

Cutting-Edge Research in SSDs

Dr. Jisung Park
Prof. Onur Mutlu
ETH Zürich
Fall 2022
04 January 2023

Pipelined & Adaptive Read-Retry

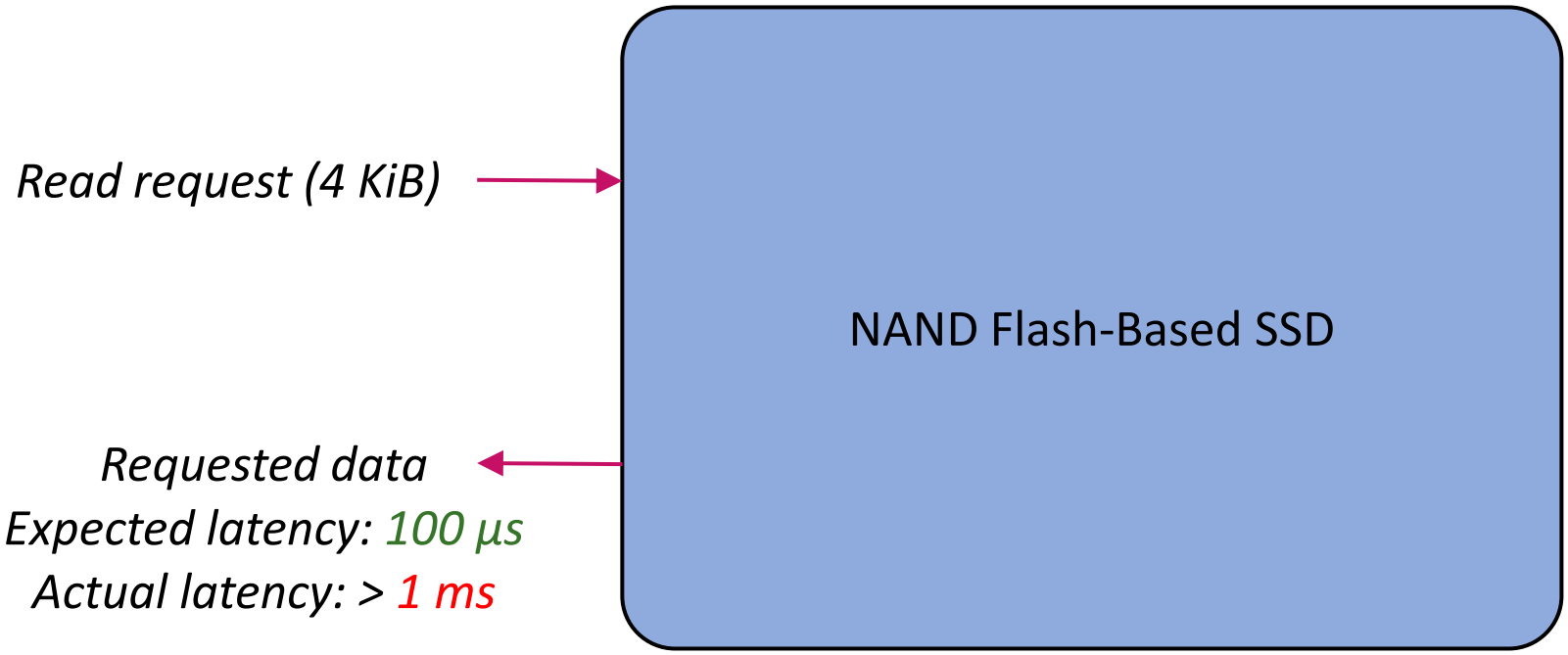
Reducing Solid-State Drive Read Latency by Optimizing Read-Retry

Jisung Park[1] Myungsuk Kim[2,3] Myoungjun Chun[2] Lois Orosa[1] Jihong Kim[2] Onur Mutlu[1]

[1]ETH Zürich Switzerland
[2]Seoul National University Republic of Korea
[3]Kyungpook National University Republic of Korea

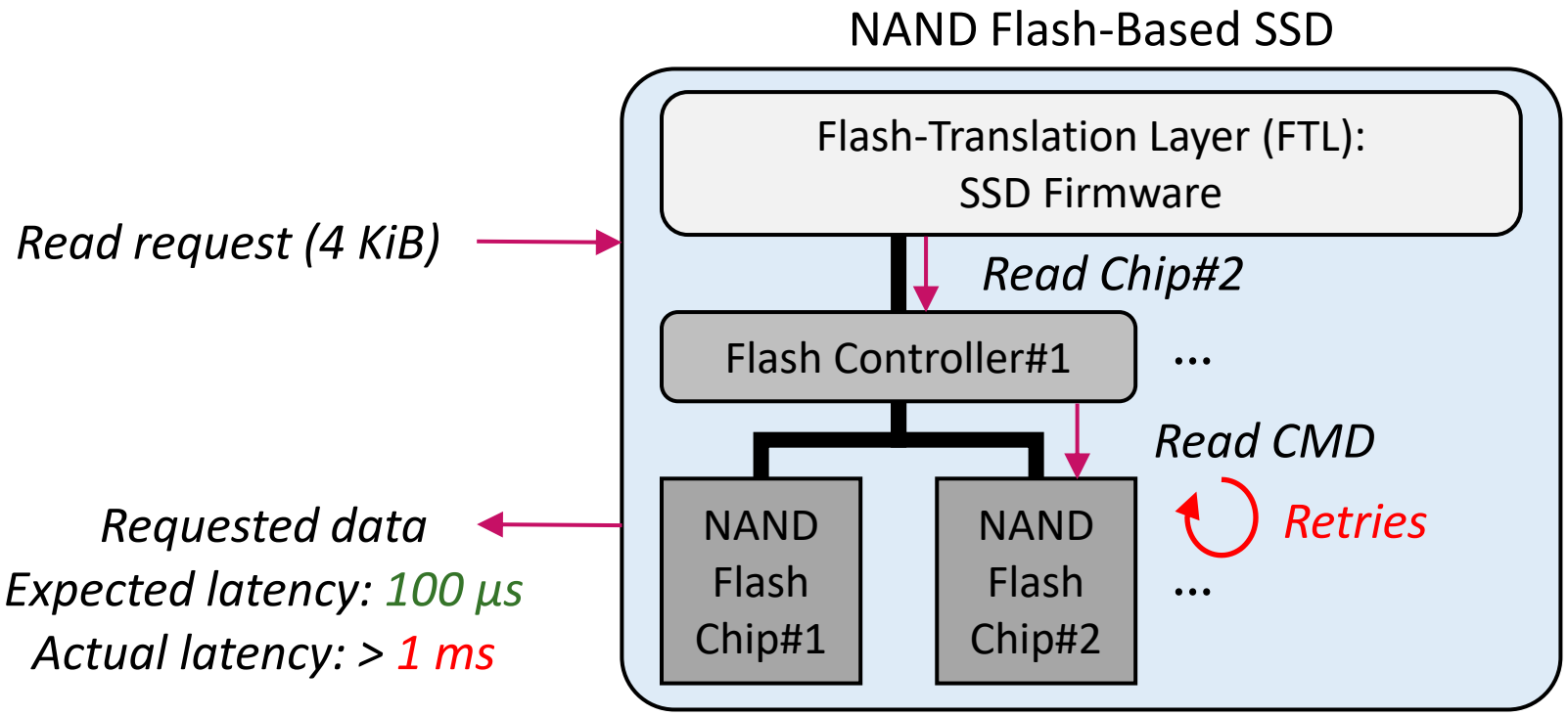
The 26th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS'21)

Problem: Long, Non-Deterministic Latency



Degrades the quality of service
of read-intensive, latency-sensitive applications

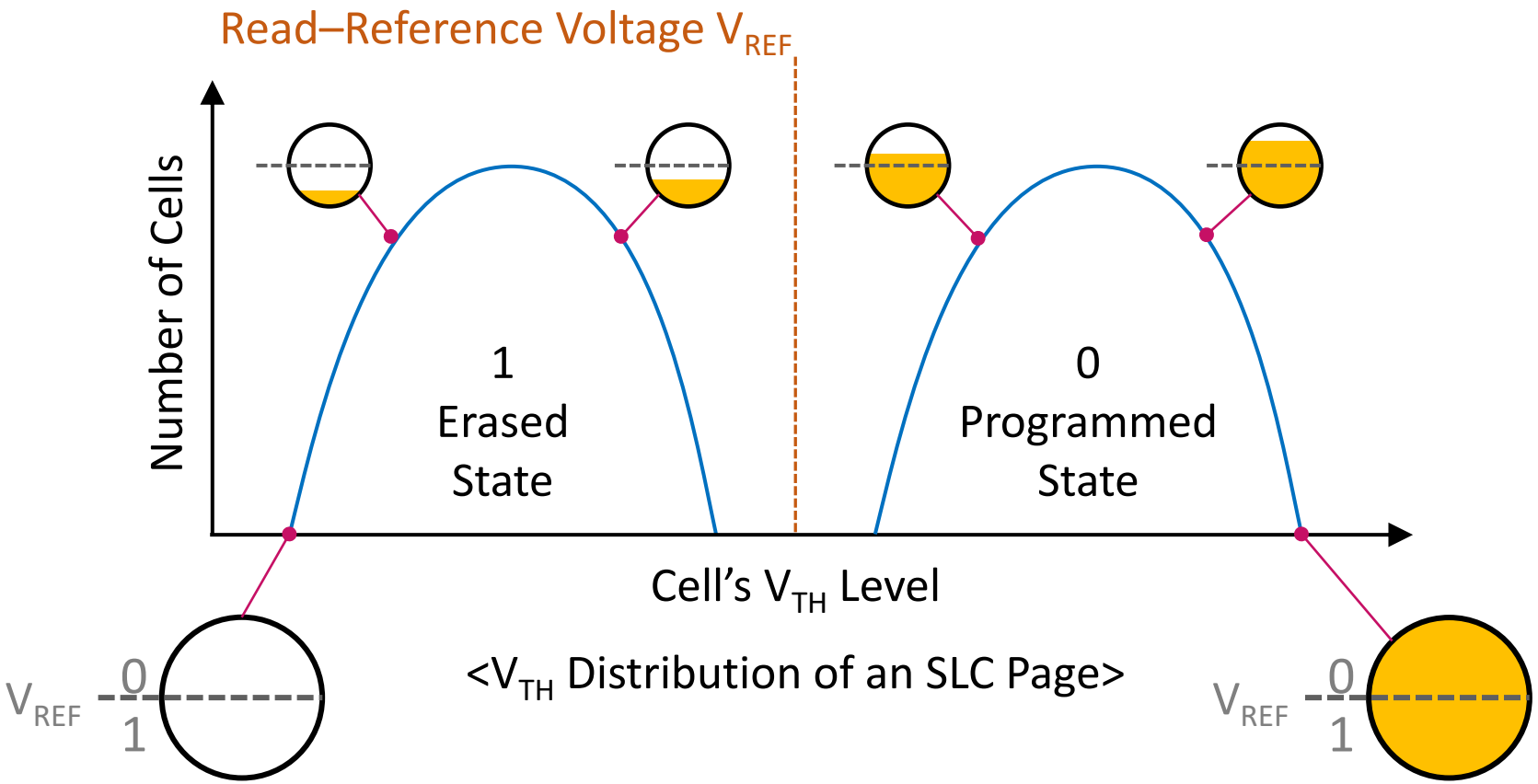
Problem: Long, Non-Deterministic Latency



Internal Read-Retry Operations

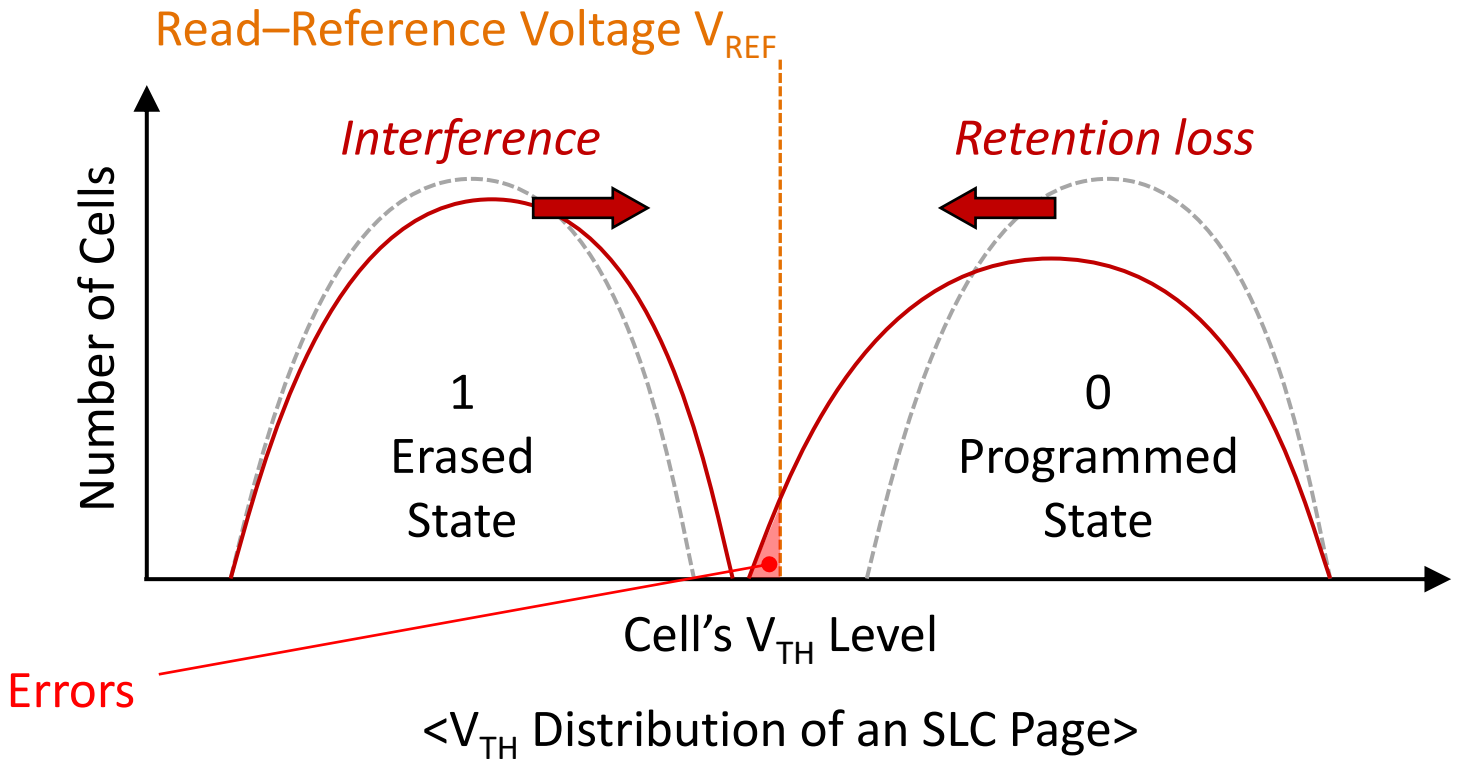
Errors in NAND Flash Memory

- NAND flash memory stores data by using cells' V_{TH} levels



Errors in NAND Flash Memory

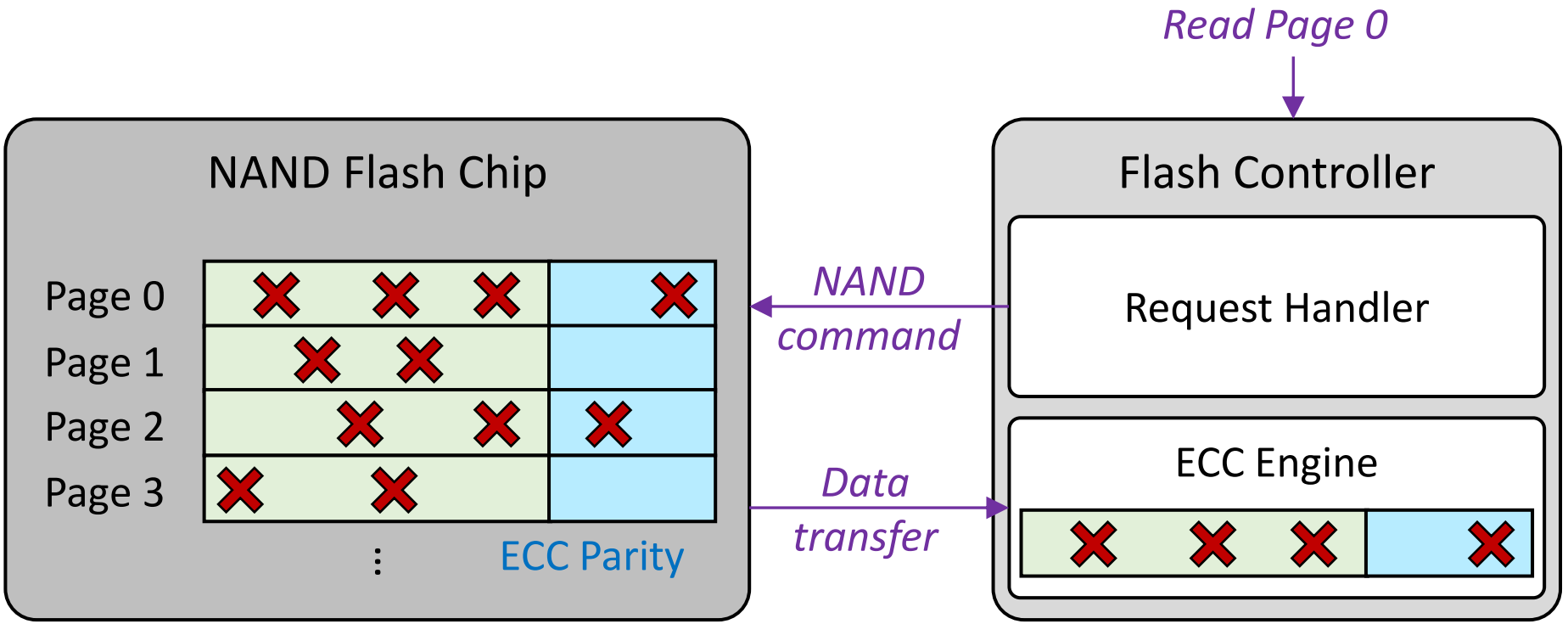
- Various sources shift and widen programmed V_{TH} states
  - Retention loss, program interference, read disturbance, etc.



<V_{TH} Distribution of an SLC Page>

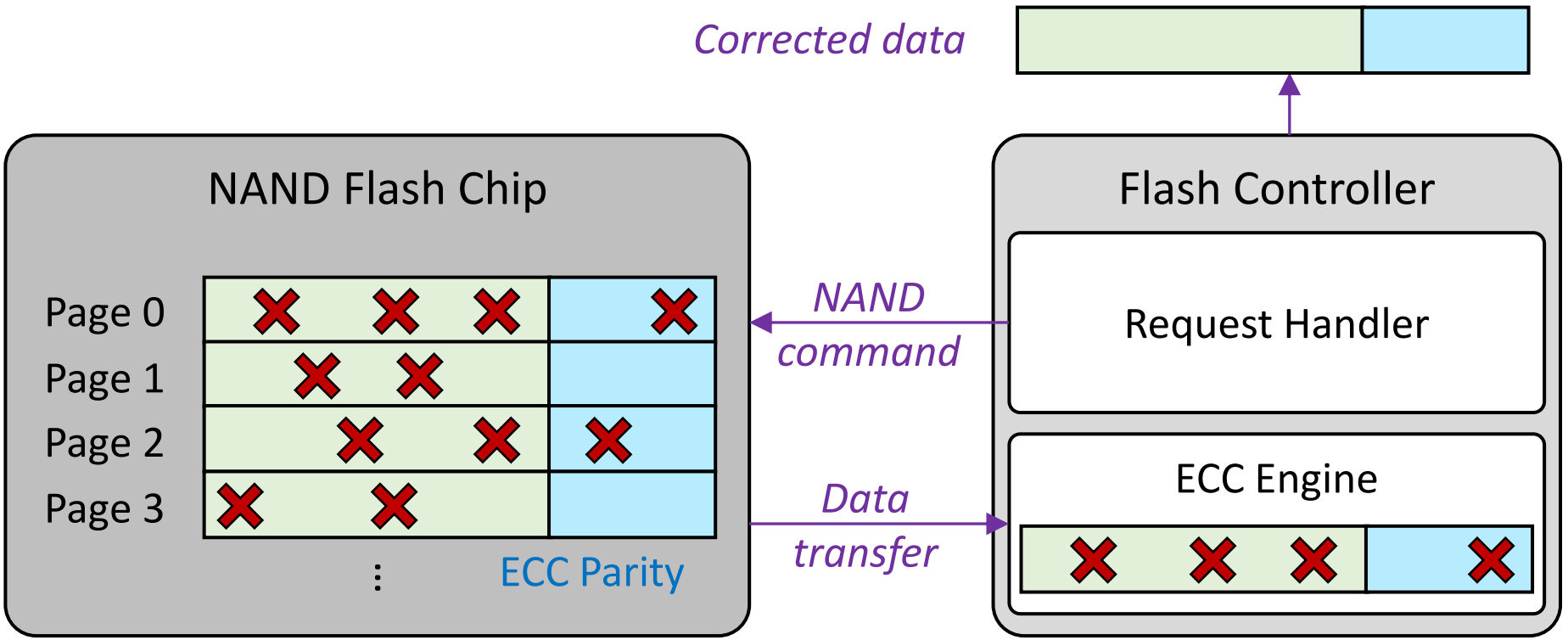
Error-Correcting Codes (ECC)

- Store redundant information (ECC parity)
  - To detect and correct row bit errors



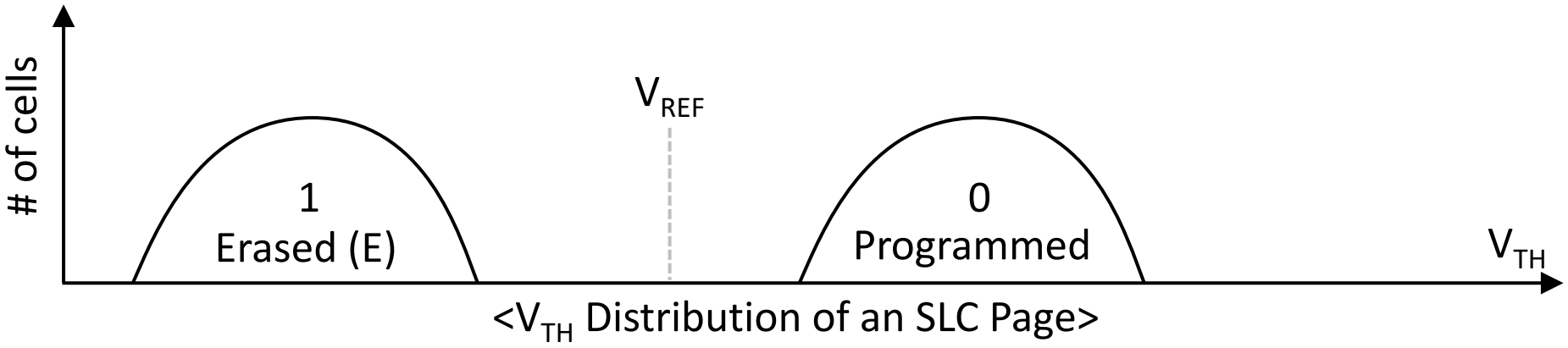
Error-Correcting Codes (ECC)

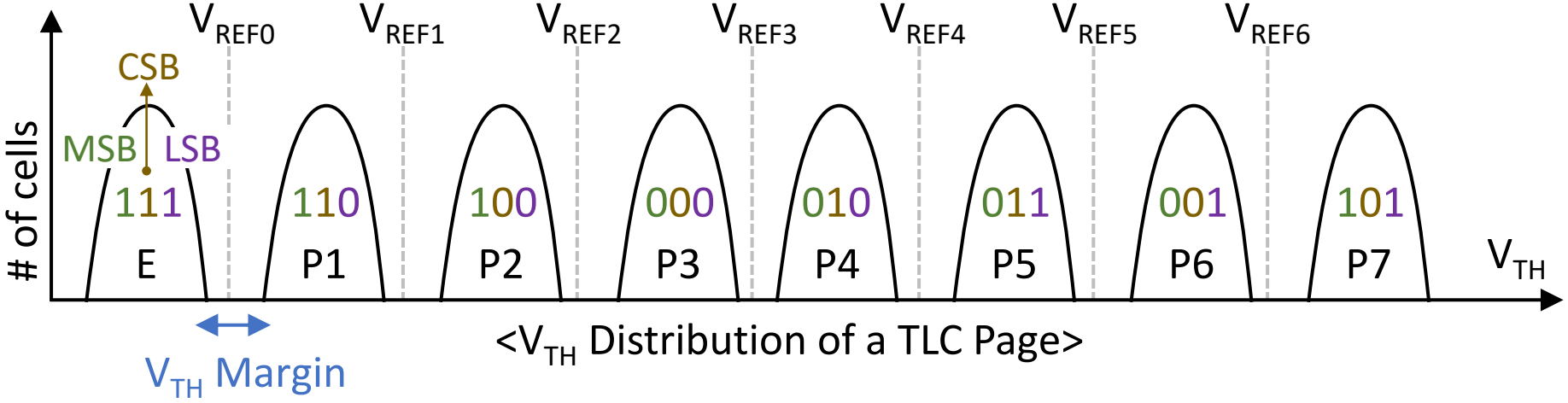
- Store redundant information (ECC parity)
  - To detect and correct row bit errors



Errors in Modern NAND Flash Memory

- High row bit-error rates (RBER) in MLC NAND flash memory
  - Narrow margin b/w adjacent V_{TH} states





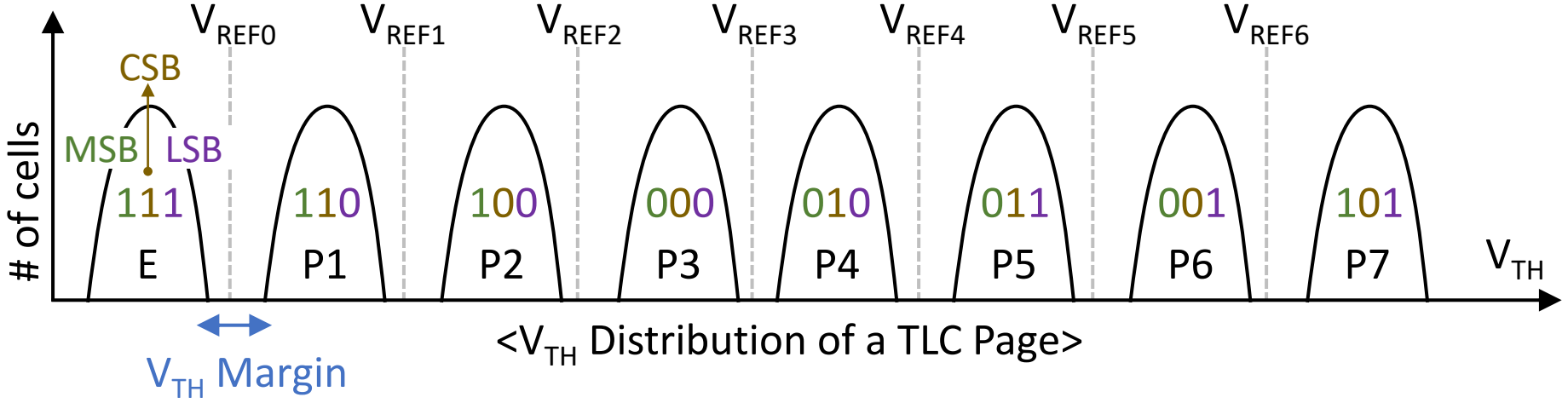
Errors in Modern NAND Flash Memory

- High row bit-error rates (RBER) in MLC NAND flash memory
  - Narrow margin b/w adjacent V_{TH} states

Strong ECC: Corrects ~80 bit errors per 1-KiB data

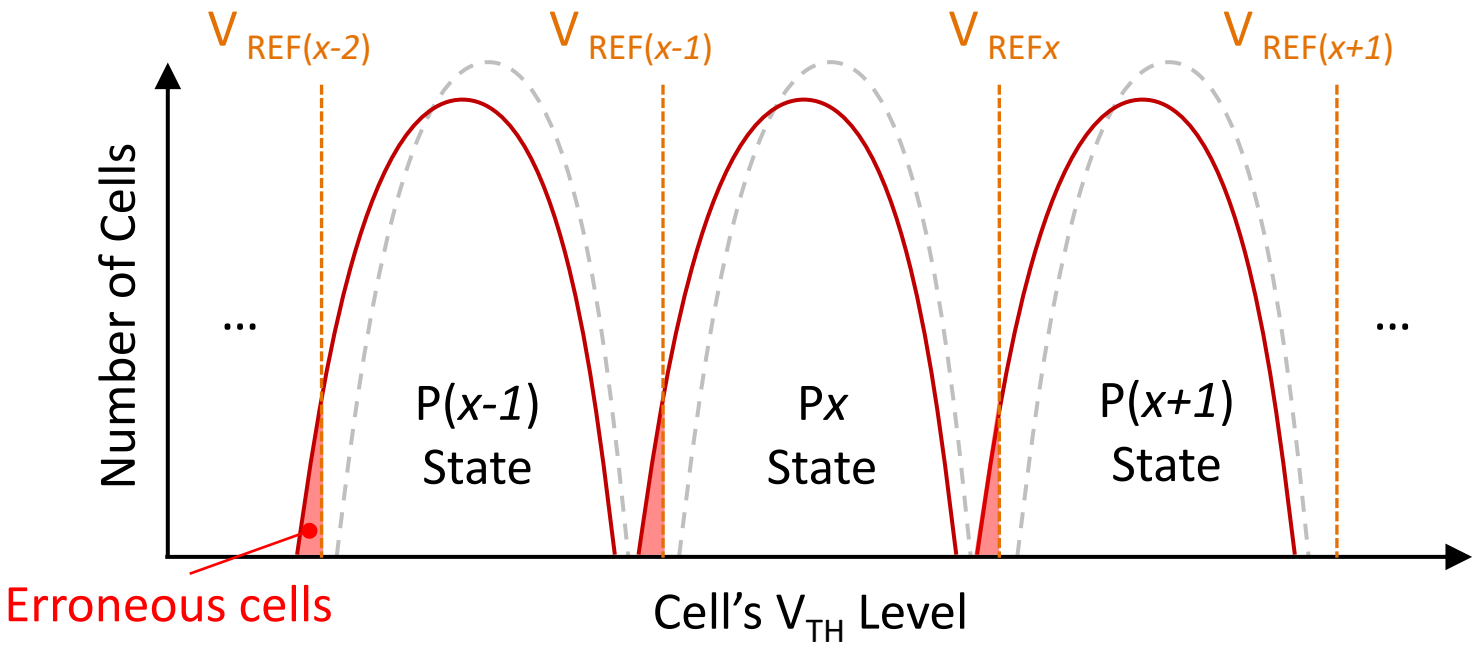
Not Scalable: Area, power, latency, …

What if RBER > ECC Capability?



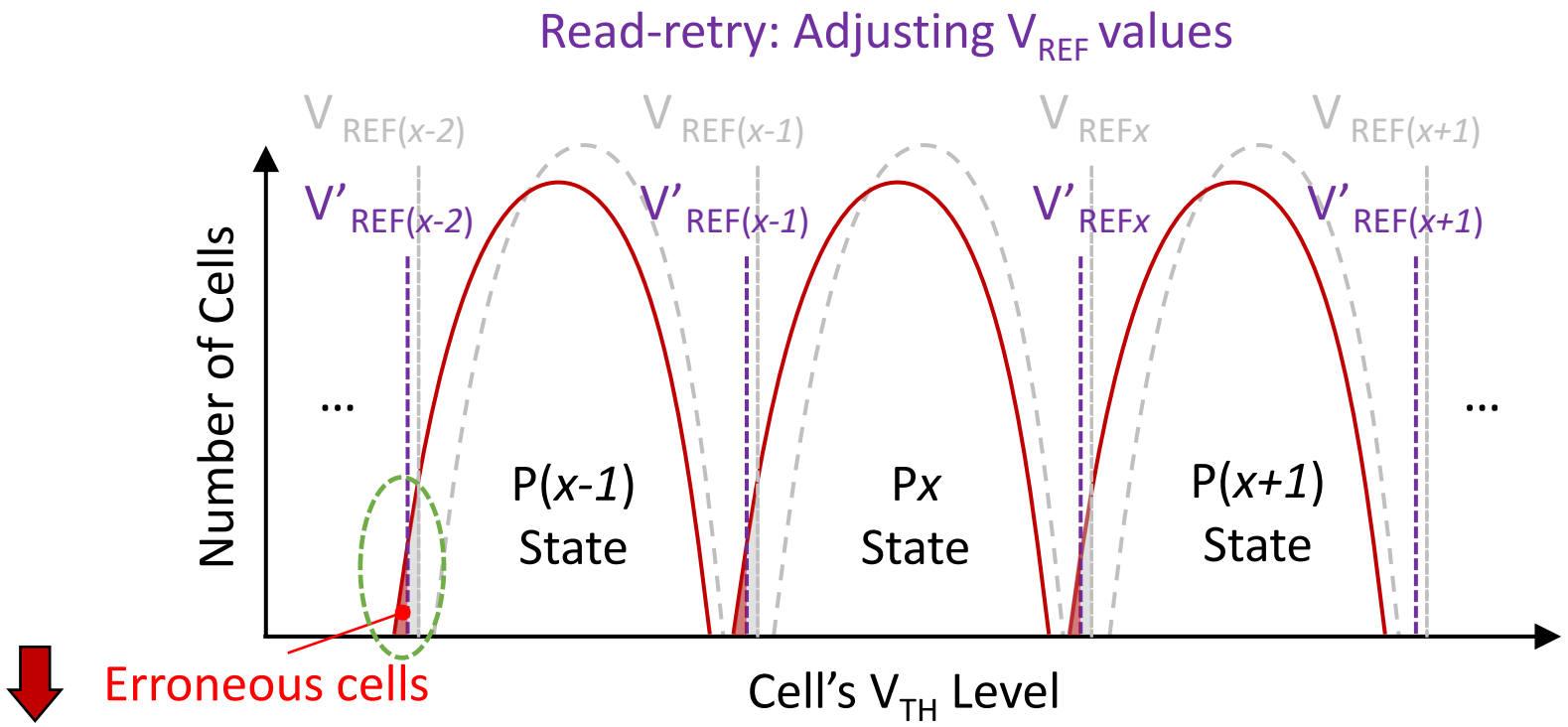
Read-Retry Operation

- Reads the page again with adjusted V_{REF} values



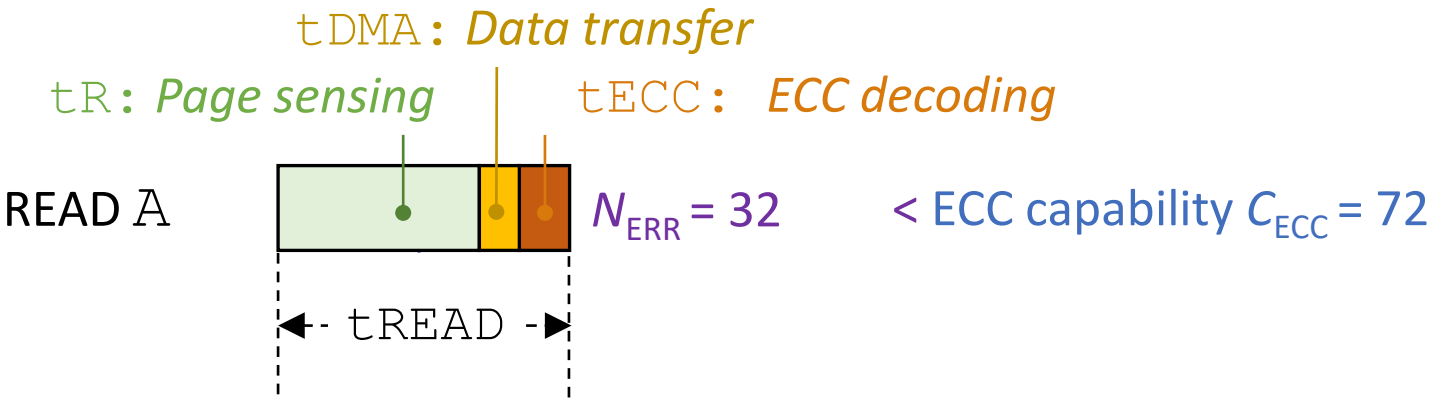
Read-Retry Operation

- Reads the page again with adjusted V_{REF} values

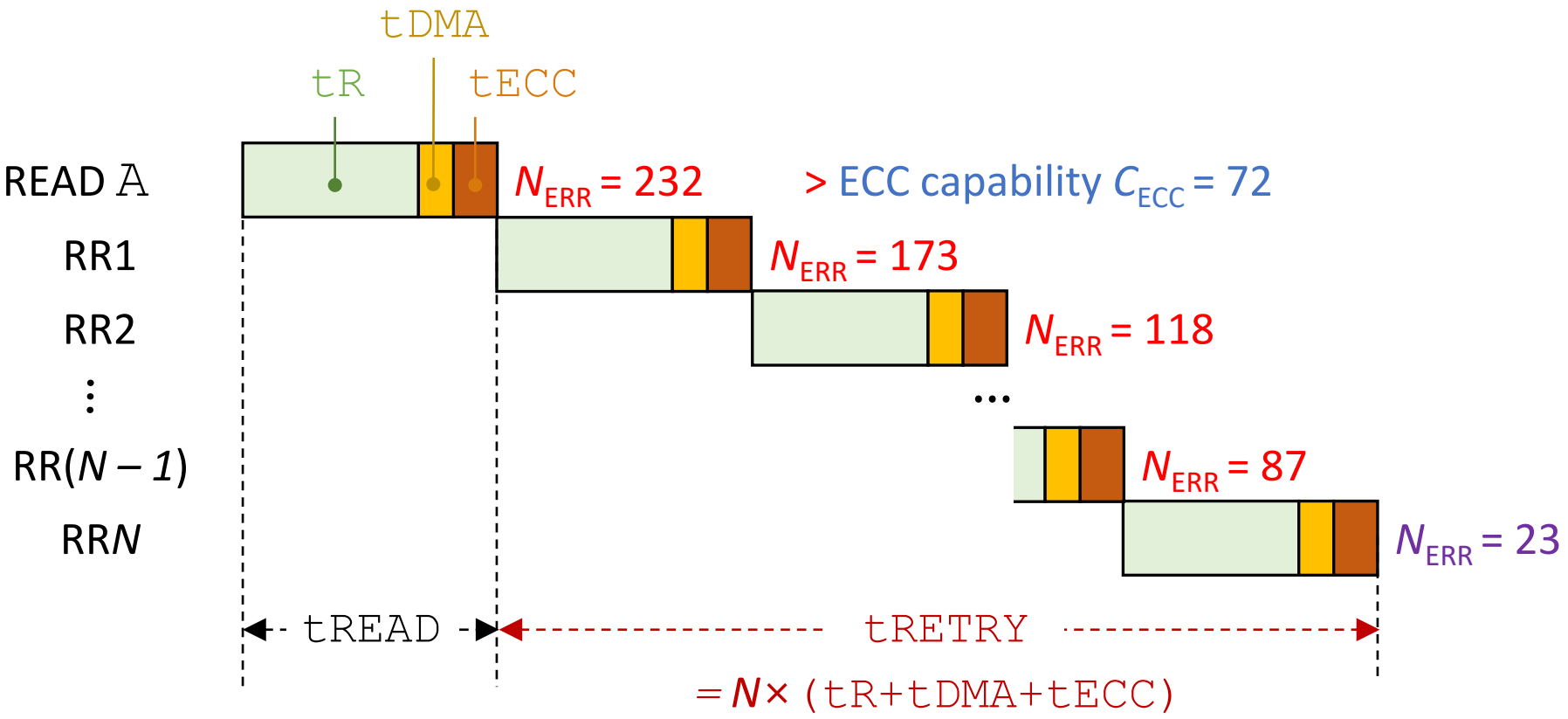


Read using properly-adjusted V_{REF} values
→ Decreases # of raw bit errors
to be lower than the ECC capability

Read-Retry: Performance Overhead



Read-Retry: Performance Overhead

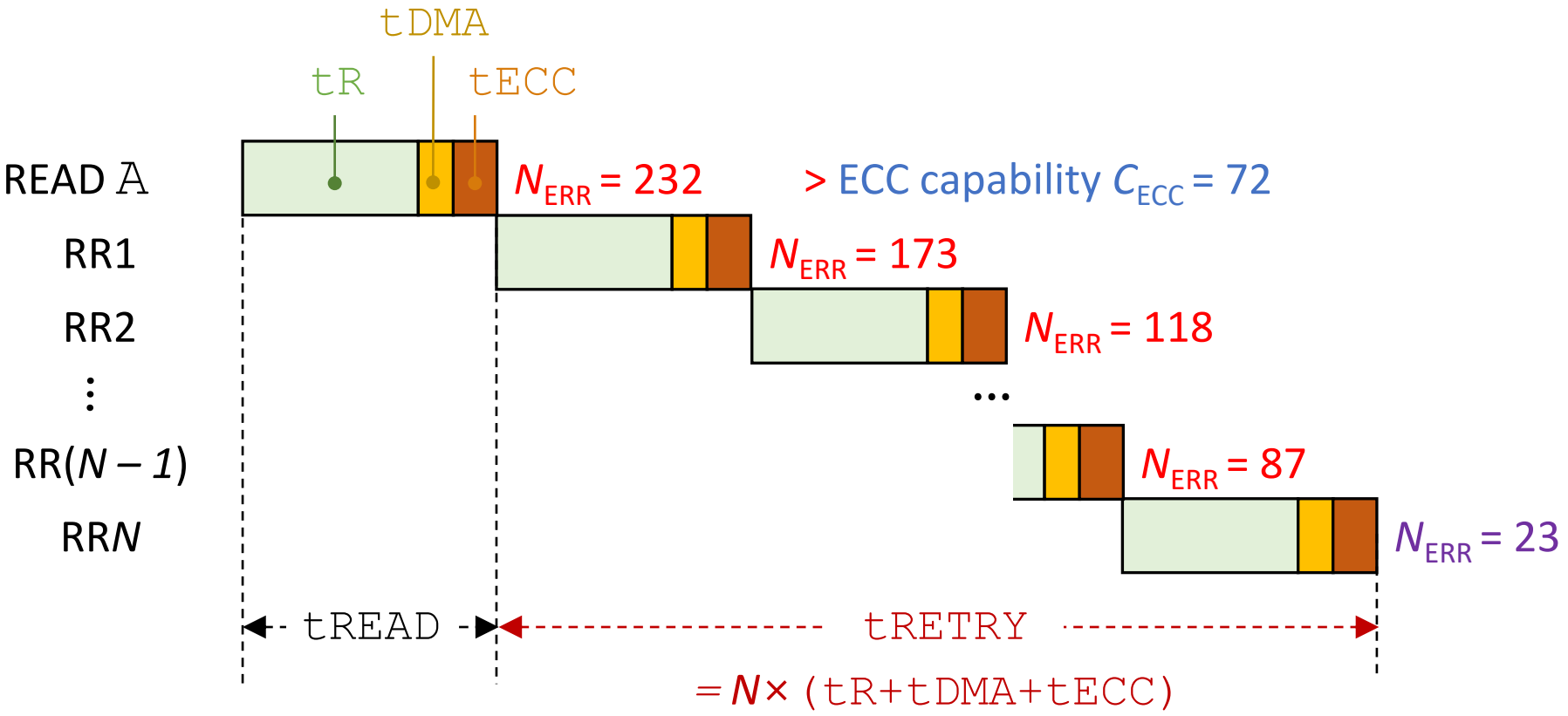


Read-retry increases the read latency
almost linearly with the number of retry steps

P&AR^2: Outline

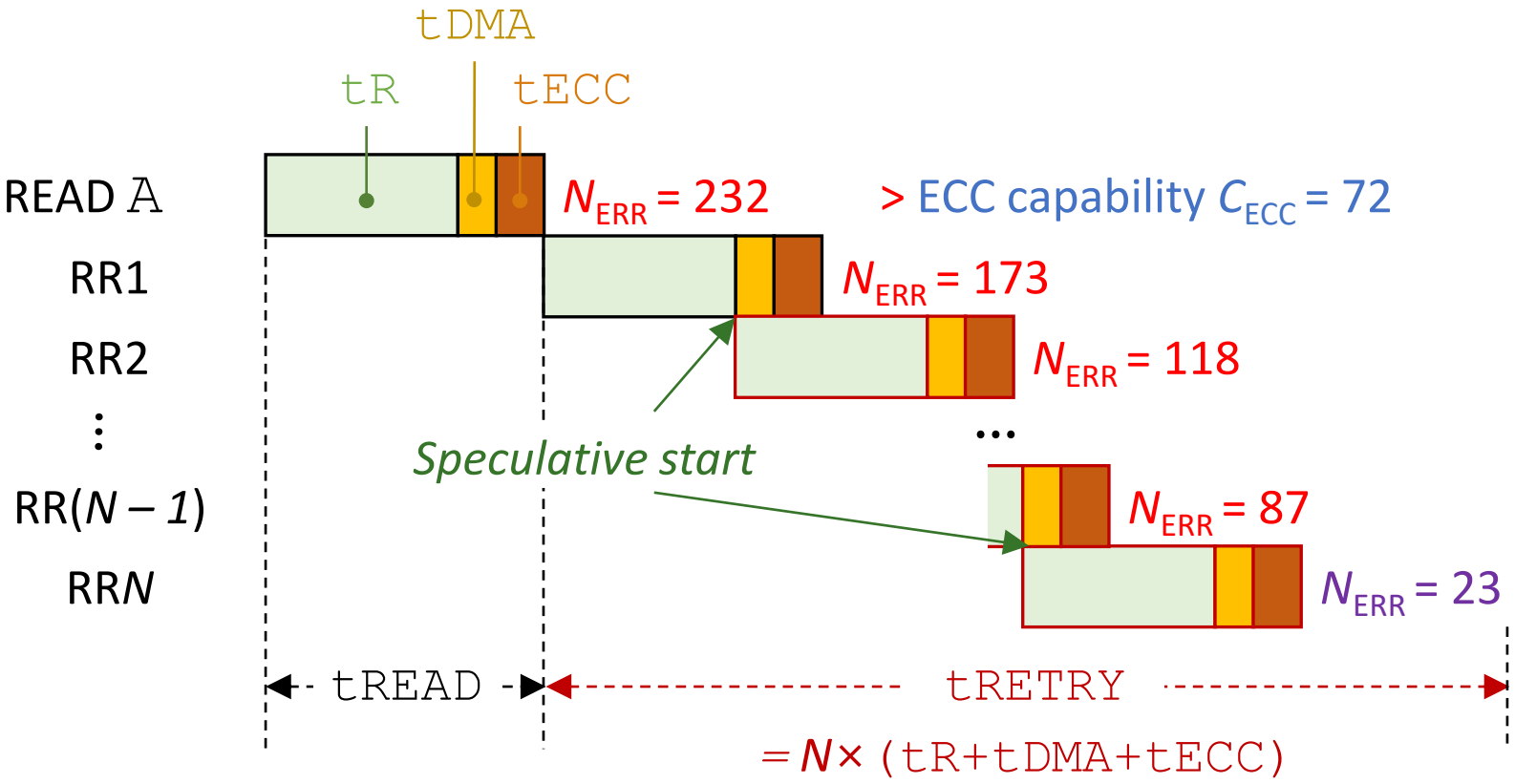
- Read-Retry in Modern NAND Flash-Based SSDs
- PR^2: Pipelined Read-Retry
- AR^2: Adaptive Read-Retry
- Evaluation Results

Pipelined Read-Retry (PR²): Key Idea



In common cases, multiple (up to 25) retry steps occur

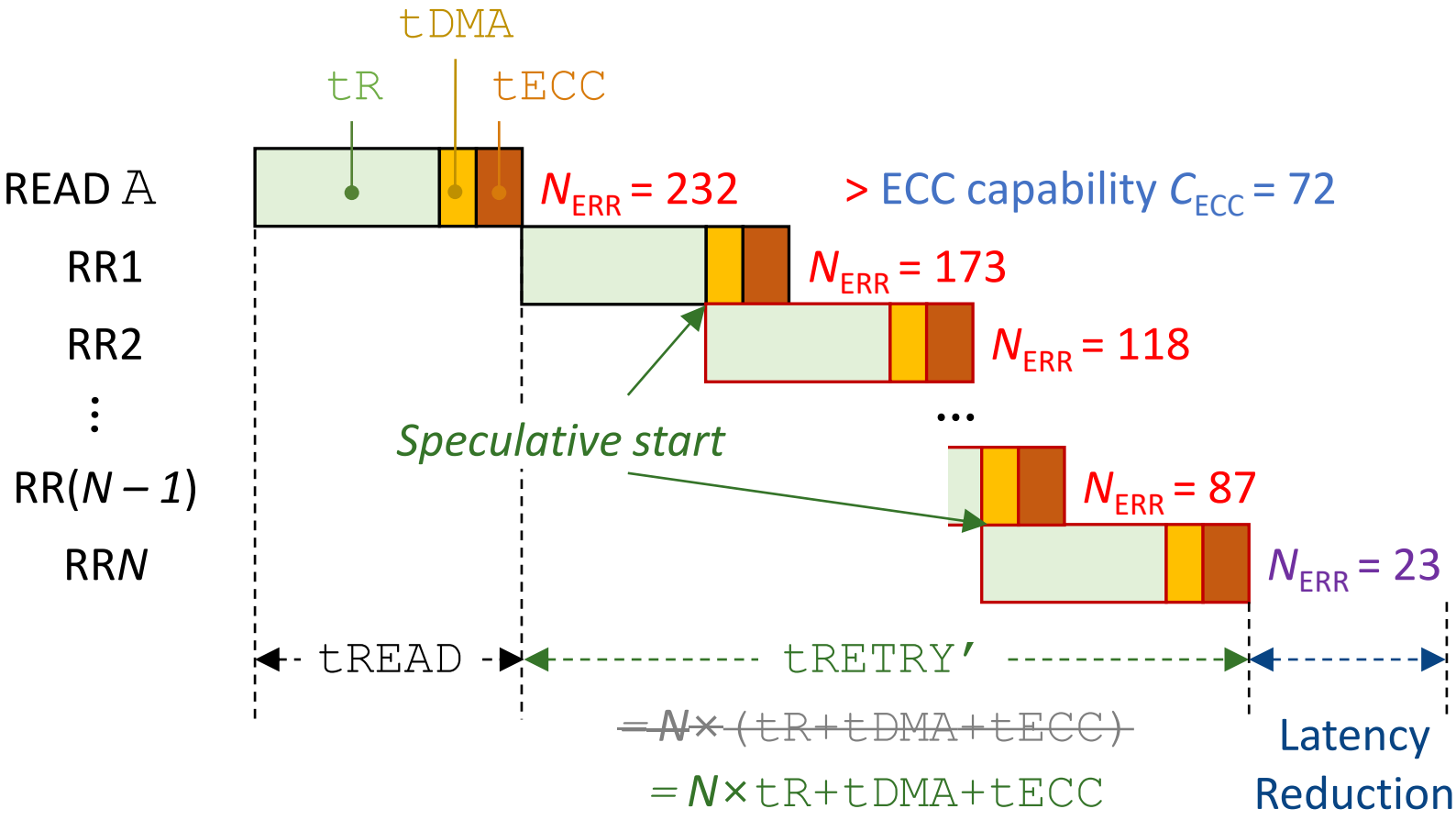
Pipelined Read-Retry (PR²): Key Idea



In common cases, multiple (up to 25) retry steps occur

→ Speculatively starts the next retry step

Pipelined Read-Retry (PR²): Key Idea



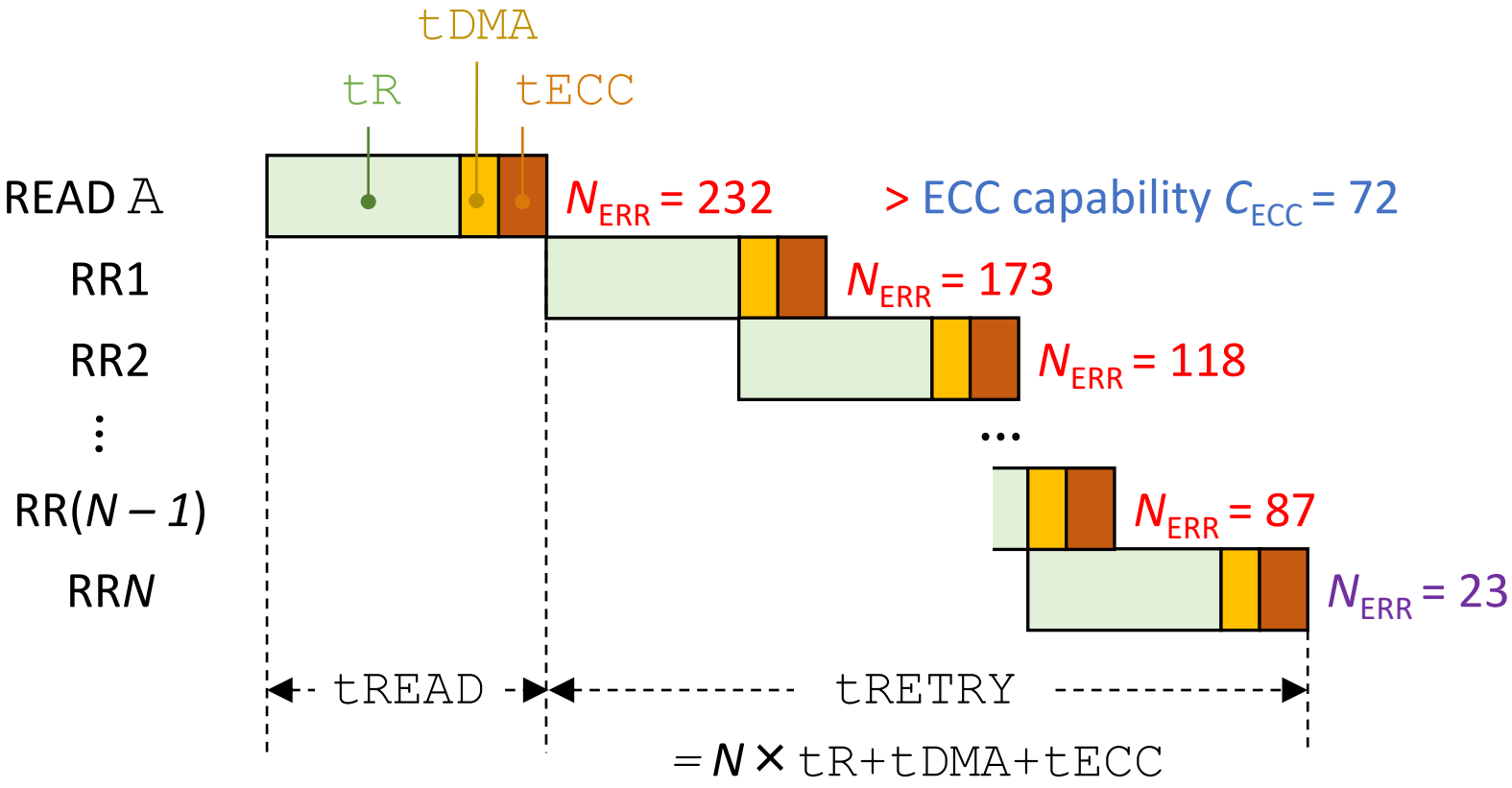
Removes tDMA & tECC

(~30% of each retry step) from the critical path

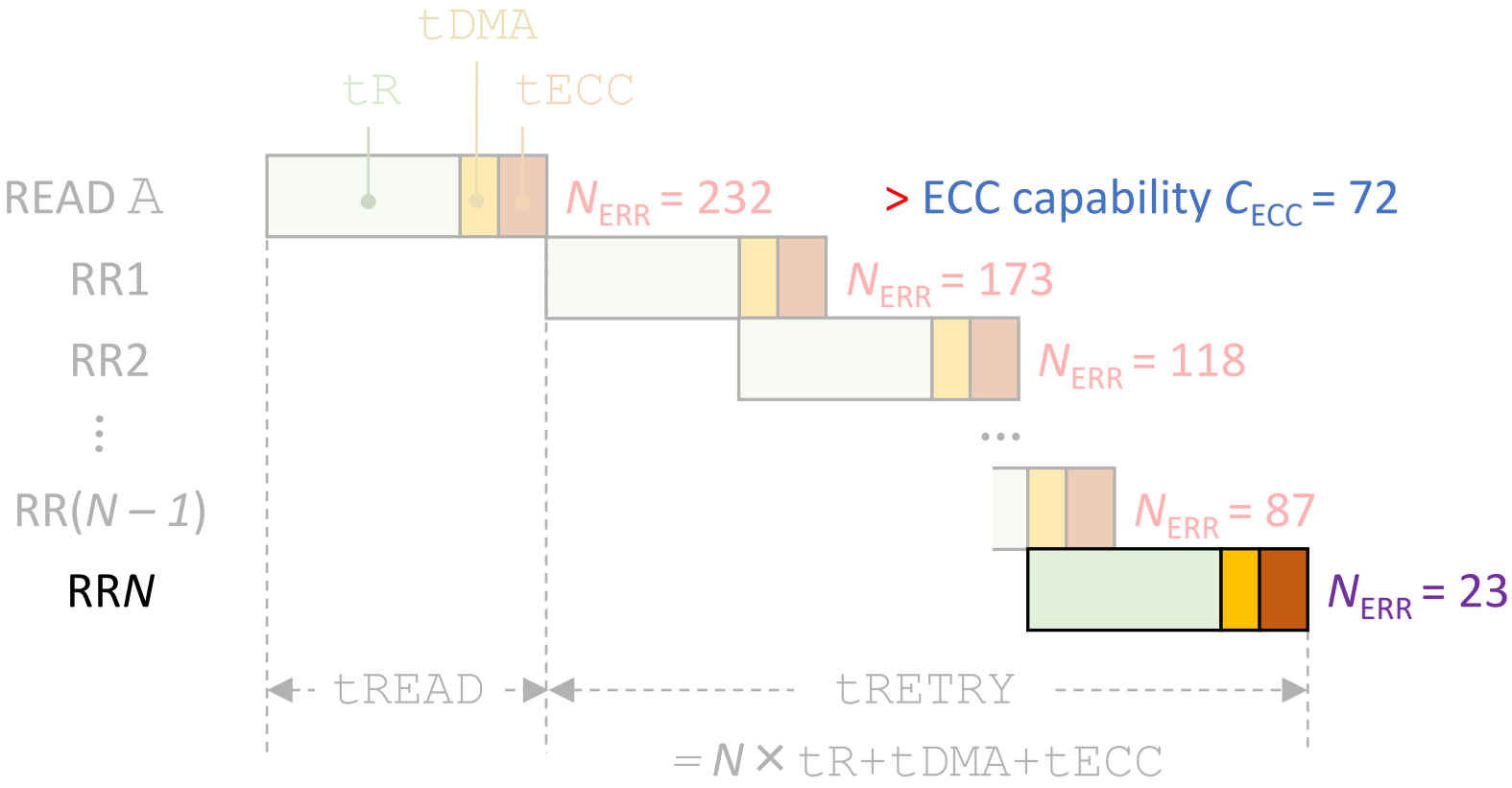
P&AR^2: Outline

- Read-Retry in Modern NAND Flash-Based SSDs
- PR^2: Pipelined Read-Retry
- AR^2: Adaptive Read-Retry
- Evaluation Results

Adaptive Read-Retry (AR²): Key Idea

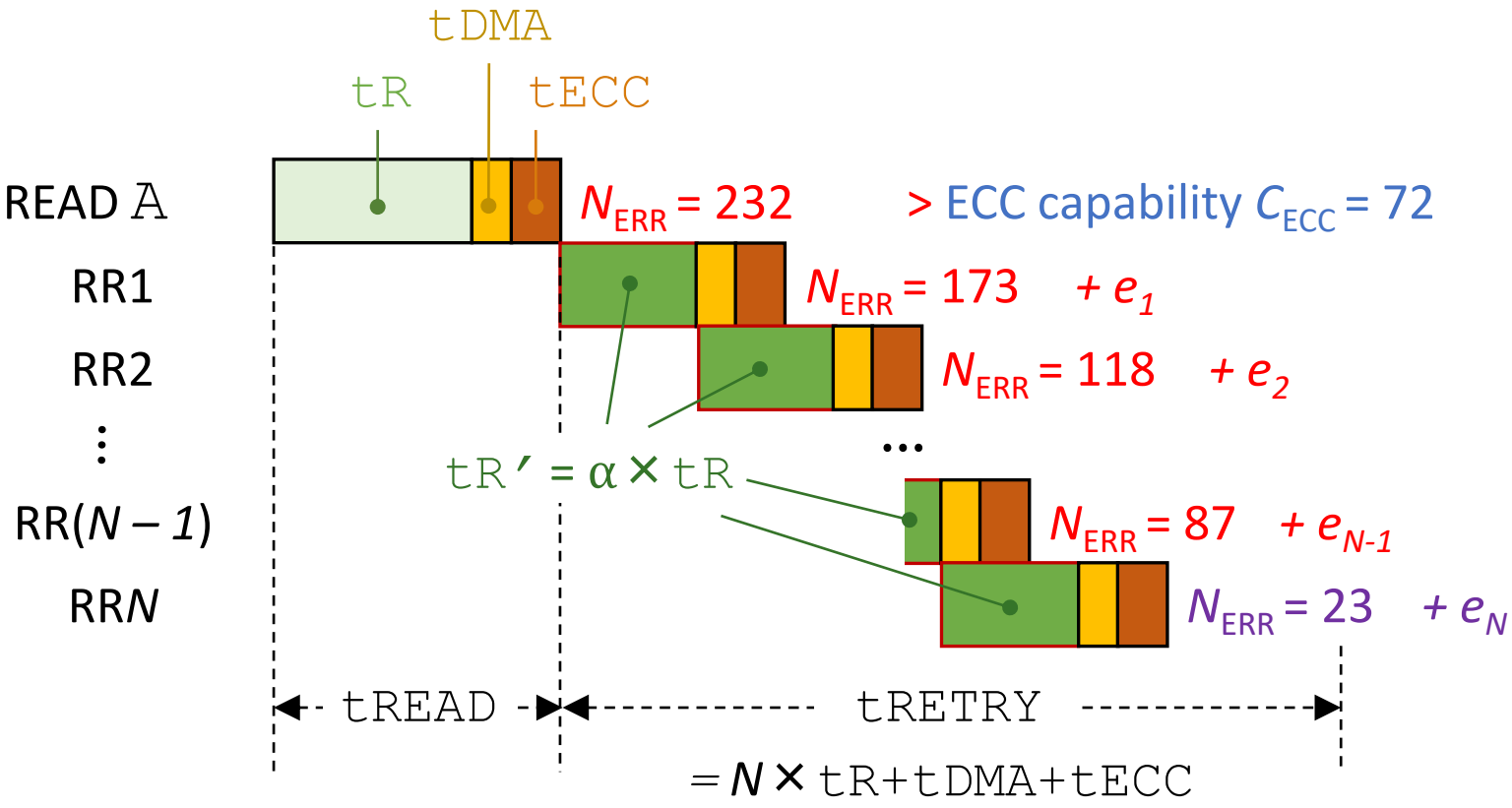


Adaptive Read-Retry (AR²): Key Idea



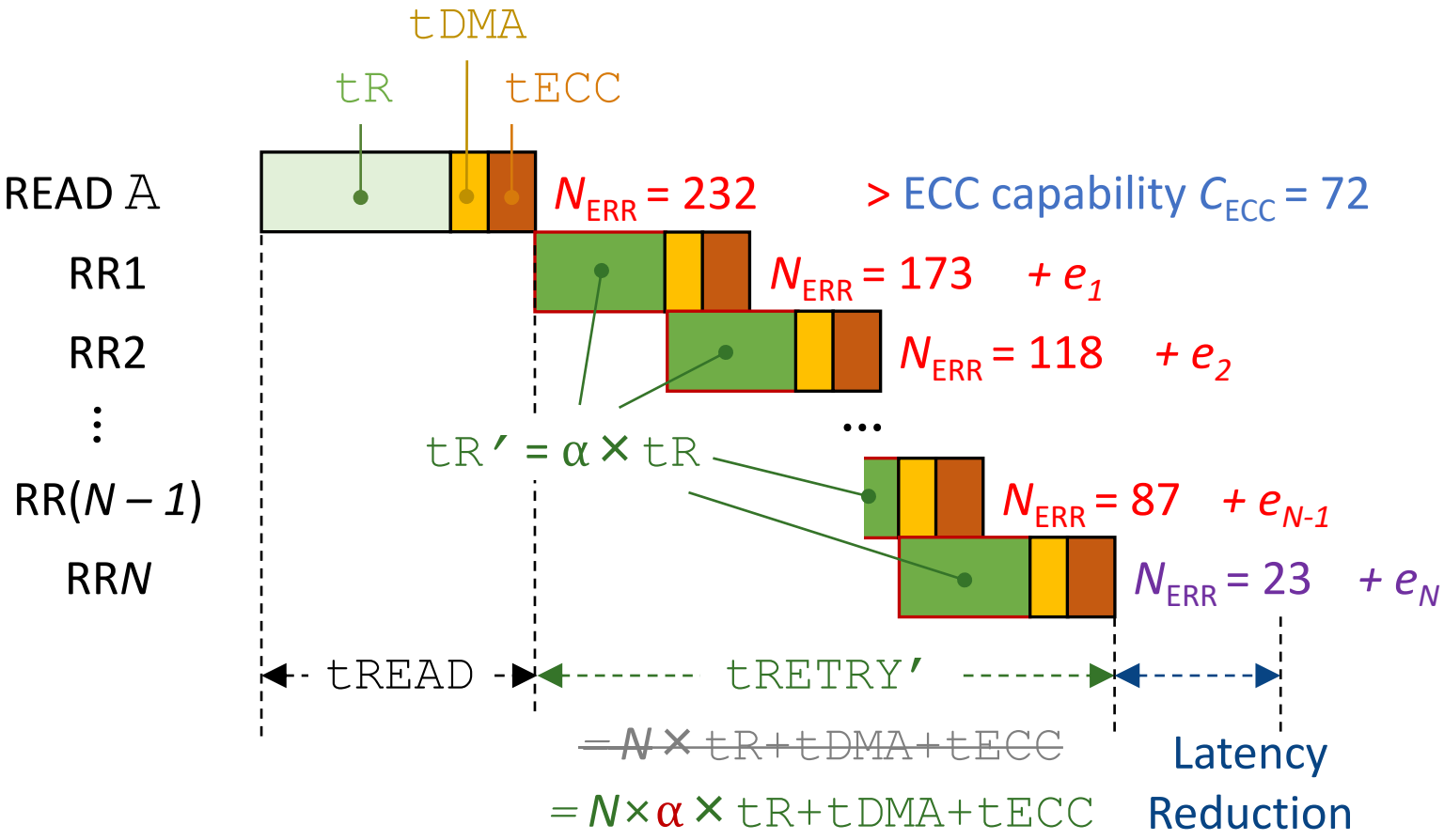
A large ECC margin
in the final retry step when read-retry succeeds
→ Can we leverage this ECC (reliability) margin?

Adaptive Read-Retry (AR²): Key Idea



Trading the large ECC margin to reduce tR

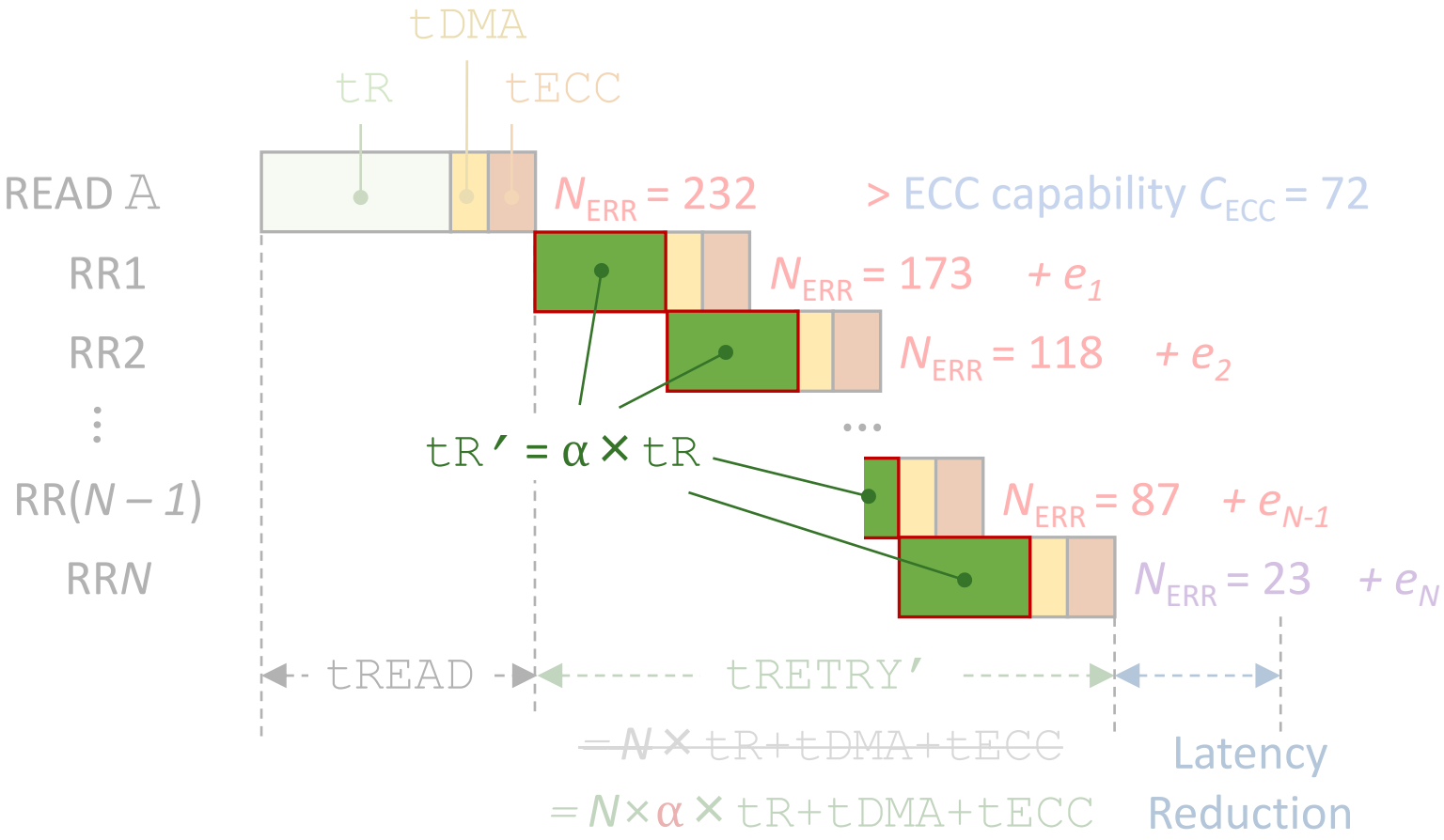
Adaptive Read-Retry (AR²): Key Idea



Trading the large ECC margin to reduce tR

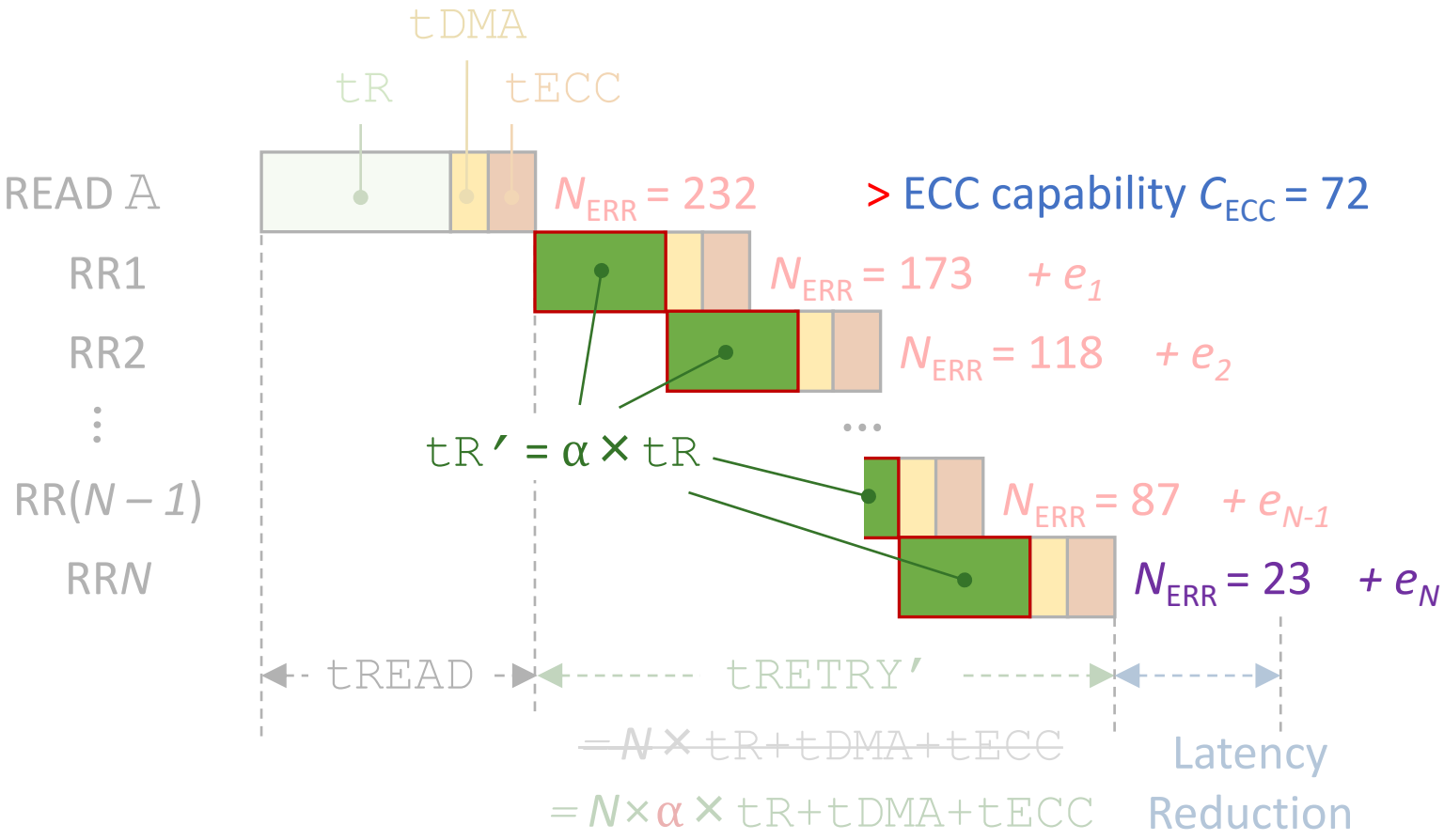
→ Further reduction in the read-retry latency

Adaptive Read-Retry (AR²): Key Idea



AR² reduces tR in every retry step

Adaptive Read-Retry (AR²): Key Idea



AR² reduces tR in every retry step

ensuring $N_{ERR} < C_{ECC}$ in the final retry step

Real-Device Characterization

- 160 real 48-layer TLC NAND flash chips

- Observation 1: A large ECC margin in the final retry step even under worst-case operating conditions
  - At most 40 errors per KiB under 1–year retention time @ 2K program and erase (P/E) cycles
  - Use of near-optimal V_{REF} in the final retry step

- Observation 2: A large reliability margin incorporated in read-timing parameters
  - 25% `tR` reduction → At most 23 additional errors
  - Worst-case-based design due to process variations

AR^2 can easily work in commodity
NAND flash chips w/ at least 25% `tR` reduction

P&AR²: Design

SSD Firmware (FTL)

P/E	t_{RET} [days]	tR [µs]
< 250	< 60	65
	⋮	⋮
	< 360	70
⋮	⋮	⋮
< 1.5K	< 60	70
	⋮	⋮
	< 360	75

Read–timing Parameter Table (Pre-profiled)

READ A

Fail

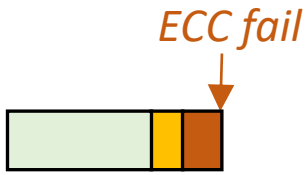
Flash Controller

ECC Engine

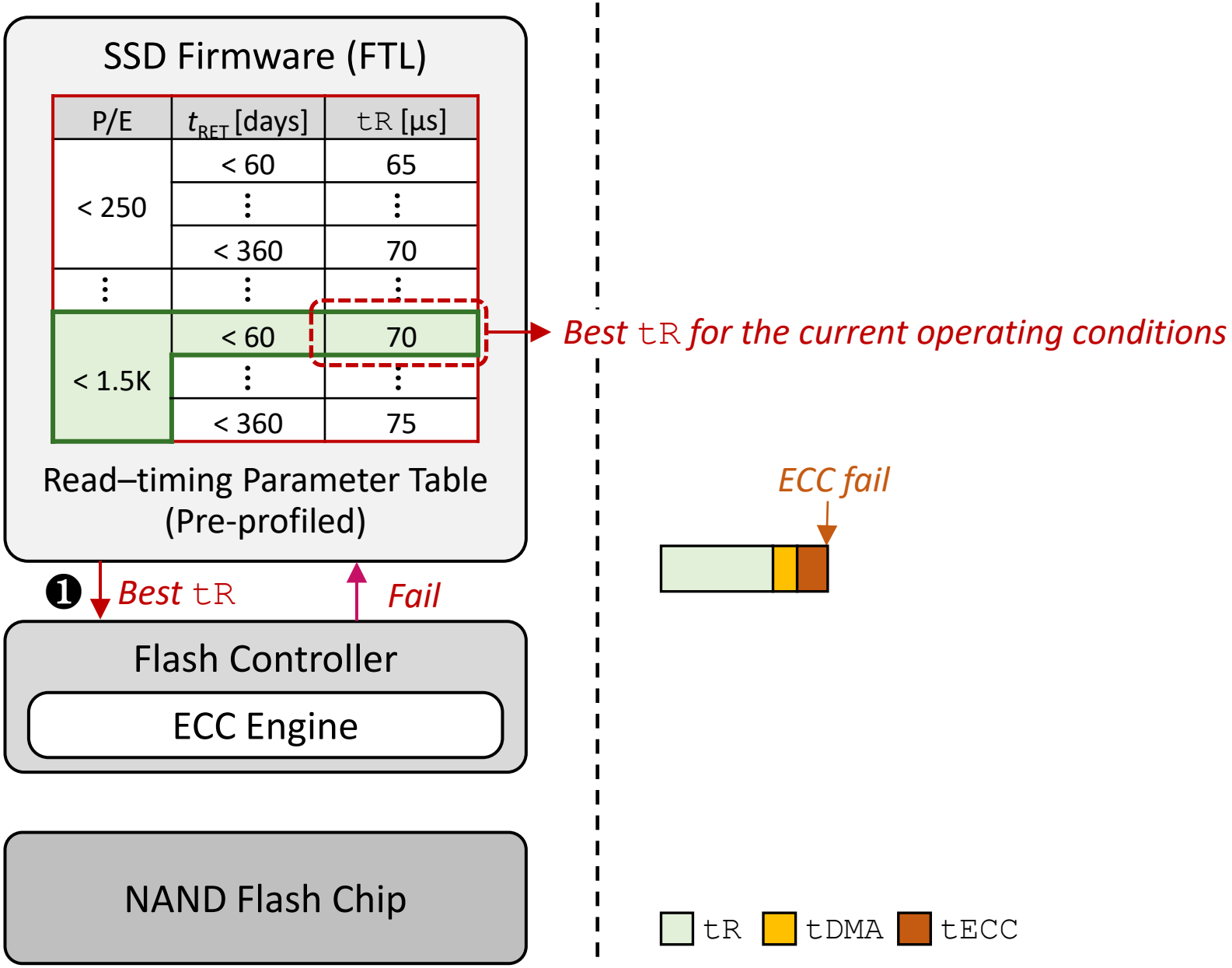
READ A

Data A

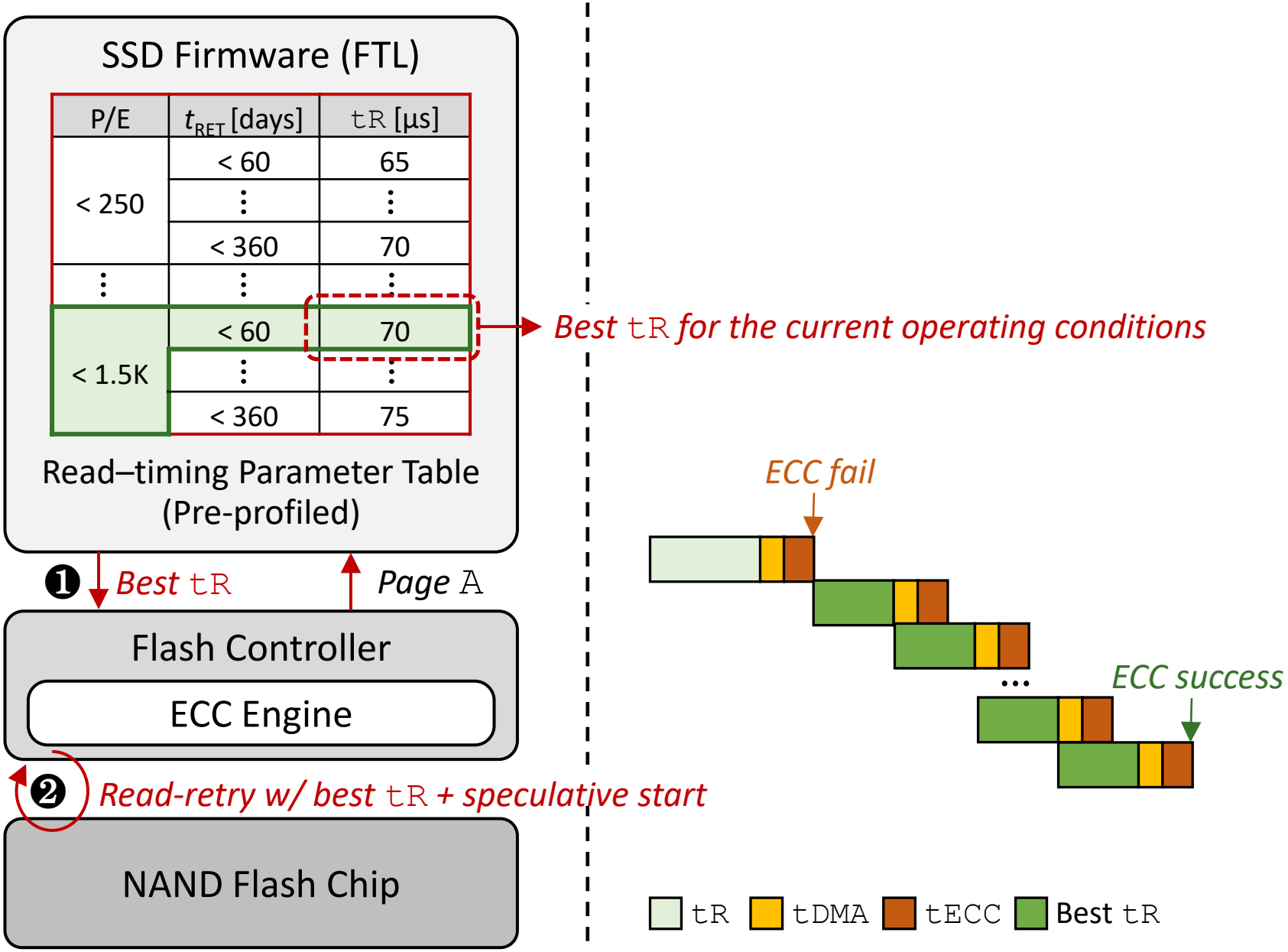
NAND Flash Chip



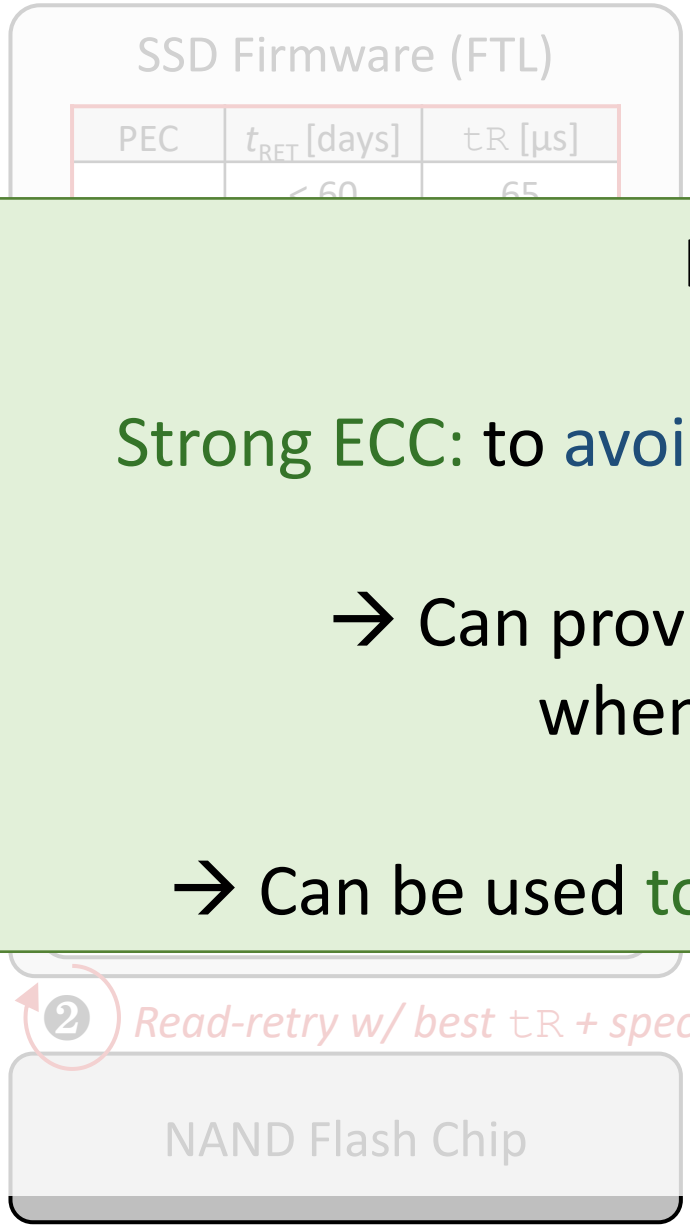
P&AR²: Design


SSD Firmware (FTL)
P/E
t_{RET} [days]
tR [µs]
< 250
< 60
65
< 360
70
< 1.5K
< 60
70
< 360
75
Read–timing Parameter Table (Pre-profiled)
Best tR for the current operating conditions
ECC fail
❶ Best tR
Fail
Flash Controller
ECC Engine
NAND Flash Chip
tR
tDMA
tECC

P&AR²: Design


SSD Firmware (FTL)
P/E
t_{RET} [days]
tR [µs]
< 250
< 60
65
< 360
70
< 1.5K
< 60
70
< 360
75
Read–timing Parameter Table (Pre-profiled)
Best tR for the current operating conditions
❶ Best tR
Page A
Flash Controller
ECC Engine
❷ Read-retry w/ best tR + speculative start
NAND Flash Chip
ECC fail
ECC success
tR
tDMA
tECC
Best tR

P&AR²: Design



Key Takeaway

Strong ECC: to avoid read-retry as much as possible

→ Can provide high reliability margin
when read-retry occurs

→ Can be used to reduce the read-retry latency

P&AR^2: Outline

- Read-Retry in Modern NAND Flash-Based SSDs
- PR^2: Pipelined Read-Retry
- AR^2: Adaptive Read-Retry
- Evaluation Results

Evaluation Results

- Simulation using MQSim [Tavakkol+, FAST18] and 12 real-world workloads

- Our proposal improves SSD response time by
  - Up to 51% (35% on average) compared to a high-end SSD w/o read-retry mitigation
  - Up to 32% (17% on average) compared to a state-of-the-art read-retry mitigation technique [Shim+, MICRO19]

P&S Modern SSDs

Cutting-Edge Research in SSDs

Dr. Jisung Park
Prof. Onur Mutlu
ETH Zürich
Fall 2022
04 January 2023